SIMULTANEOUS LOCALIZATION AND MAPPING AT SMALL BODIES COMBINING INCREMENTAL OPTIMIZATION AND SEQUENTIAL ESTIMATION

F. Piccolo[1], P. Califano[2], P. Panicucci[3], and F. Topputo[4]

[1]*Politecnico di Milano, Milan, Italy, felice.piccolo@polimi.it*
[2]*Politecnico di Milano, Milan, Italy, pietro.califano@polimi.it*
[3]*Politecnico di Milano, Milan, Italy, paolo.panicucci@polimi.it*
[4]*Politecnico di Milano, Milan, Italy, francesco.topputo@polimi.it*

ABSTRACT

A set of autonomous navigation algorithms for small-body exploration is proposed, coupling pose-only factor-graph based SLAM with an EKF for velocity estimation. Monocular camera, star tracker, and LIDAR altimeter measurements are tied together in a unified framework. Rotational states are estimated using light-curves and the DAAVE algorithm, refined by a MEKF. Simulations on Bennu and camera-in-the-loop tests show estimation accuracy in the order of 10–20 m position, $<0.5°$ attitude, and 0.1 mm/s velocity errors. Loop closures and LIDAR measurements are found to be crucial in bounding the uncertainty of the estimates. The proposed design reduces the graph complexity with respect to a fully coupled implementation and supports staged, autonomous approach operations.

Key words: autonomous small bodies characterization; monocular SLAM; factor graph; lidar altimeter.

1. INTRODUCTION

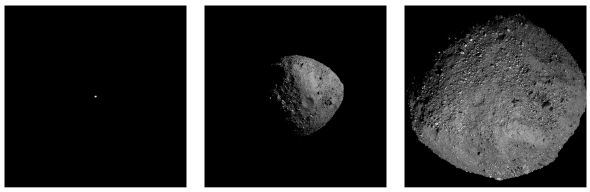
Recent deep-space missions have revealed the wide diversity of asteroids in size, shape, rotation, and surface properties. However, only preliminary information can be inferred from Earth. High-fidelity characterization necessarily requires in-situ data gathered by dedicated space missions. At present, small body properties are estimated on ground: images are downlinked from the spacecraft and processed in batch to recover both the body properties and the spacecraft state [1, 2]. An essential tool currently used in this process is stereophotoclinometry (SPC) [3], which is however compute-intensive, needs manual intervention to foster convergence, and drives downlink volumes due to the required high number of images. Increased on-board autonomy can shift part of this workload to the spacecraft, reducing ground dependence and data exchange while improving responsiveness and science return.

Simultaneous localization and mapping (SLAM) offers an alternative to the current ground-centric paradigm by estimating the spacecraft pose and a consistent map of the target online, and has gained attention for small-body exploration [4–6]. Recent works have extended SLAM toward full target characterization—spin axis, rotational period, inertia, mass, and gravity [6–10]. Most of these systems adopt either a graph-based or a filtering approach. A first exception investigating a hybrid graph-filter design has been shown in a LIDAR-centric pipeline [5]. Later, a preliminary comparison of graph-filter approaches was presented in [11], indicating that delegating dynamics to a filter can still preserve accuracy while lowering factor-graph complexity. Many monocular methods assume known scale or use a single depth anchor, whereas removing the scale ambiguity inherent of the SLAM problem with explicit LIDAR ranges is less common [10].

This work presents a SLAM algorithm that combines a factor graph for spacecraft pose estimation and mapping with an EKF to estimate velocity states. The sensing suite assumes a monocular camera, a LIDAR altimeter, and a star tracker, where the LIDAR range measurements is realistically introduced in the estimation process to solve scale ambiguity. This mix is common on deep-space probes and is suitable even for small platforms [12]. Two additional dedicated routines estimate the rotational period and spin axis. Following typical approach strategies [13], the rotational period is estimated at larger distances, then refined together with the spin axis when the target is resolved but still within the camera FOV. Finally, the full SLAM pipeline is run in close proximity. This staged estimation supports realistic and safe operations before committing to closer orbits, which require a high degree of target knowledge accuracy.

2. SLAM ALGORITHM

The SLAM algorithm is based on an estimation back-end that relies on incremental factor graph optimization using iSAM2 [14], and an image processing front-end consisting in a keyframe-based pyramidal implementation of the KLT optical tracker [15]. An EKF is run in parallel to iSAM2 in order to estimate the spacecraft velocity, which is not included in the factor graph. The following sections describe these two components in detail.



(a) Far. (b) Intermediate. (c) Close.

Figure 1: Representative images from each of the three phases. At **far** distance, the rotational period is estimated. At **intermediate** distance, the spin axis is determined and the angular velocity is refined. Finally, the full SLAM algorithm is run in the **close proximity** phase.

2.1. Image processing

The front-end module is based on a Pyramidal KLT algorithm. The tracking approach has been preferred over feature matching because limited appearance changes are expected between consecutive images, given the relatively slow dynamics that characterize spacecraft in proximity of small bodies. Indeed, typical velocities (relative to the target) are in the order of cm/s or dm/s. Illumination changes play a role for long-term tracking, since shadows on the surface change significantly as the small body rotates. Nonetheless, image-to-image changes are small, and appropriate feature selection limits the impact of shadows on the tracking accuracy.

Features are interest points in the image that can be recognized in subsequent frames. A variety of feature detection algorithm are available in the literature. In this work, the Harris detector is used, a well-known and computationally affordable algorithm [16]. After the initial set of features is extracted, two postprocessing steps are applied. First, features close to the asteroid outline and the terminator region are discarded. These are known to provide poor tracking performance [17]. Second, a clustering procedure is applied to select a subset of features spread over the image plane. The k-means algorithm is used to cluster features. Then, for each cluster the feature with the strongest corner response is stored as the corner response is a measure of the feature quality [16].

When performing feature tracking on an image sequence, keypoints are periodically detected in a subset of frames, called reference images or keyframes. They are then tracked on subsequent frames until too many features are lost, at which point a new extraction step is performed. Keyframes are also used to reject possible outliers generated by the tracking process. At each frame, a MSAC-based outlier rejection step is carried out, which relies on the estimation of an essential matrix relating the current frame to the last keyframe.

When a new keyframe is extracted, the remaining inliers are passed to the back-end algorithm, which treats their image coordinates as measurements in the estimation process.

2.2. Factor Graph construction

The factor graph formulation represents the essential backbone of modern SLAM systems, as it allows accurate and efficient estimation through incremental optimization algorithms. In this work, the framework is exploited to combine observations of surface features, provided by the front-end, with LIDAR range measurements.

Feature observations in multiple images are encoded using projection factors, which account for the reprojection error of each landmark. When a new keyframe is inserted, the currently active features are divided in two sets. The first contains those features that already have an associated landmark in the graph. For each feature in this set, a new projection factor ϕ_{proj} is added. The factor's noise model is constructed based on the track length, that is the number of consecutive images the same feature has been observed. For a track length l_T (i.e., number of frames), the standard deviation of the projection factor is modelled as $\sigma_{\text{proj}} = 0.1 l_T$. This is to account for the drifting behaviour typical of feature tracking algorithms. Finally, the covariance matrix is assumed to be isotropic.

The second set of features contains new observations and landmarks that had already been observed, but have not yet been inserted into the graph. These are stored until a minimum number of observations n_{init} is available, after which the landmark is triangulated and initialized. This value was set to $n_{\text{init}} = 3$ for this work. The triangulation is based on a multi-view DLT algorithm [18], minimizing the feature reprojection error across all keyframe views [19]. Once a landmark is initialized, projection factors are added for all keyframes in which it was observed, using the same noise model discussed above. After every feature has been processed, the SLAM solution is incrementally updated using iSAM2 in two separate steps. The first is carried out after the introduction of projection factors for the already observed landmarks (set 1), while the second after the initialization step (set 2). This strategy has been found to increase the robustness of the optimization process with respect to performing a single update including all newly added factors. As the problem is non-

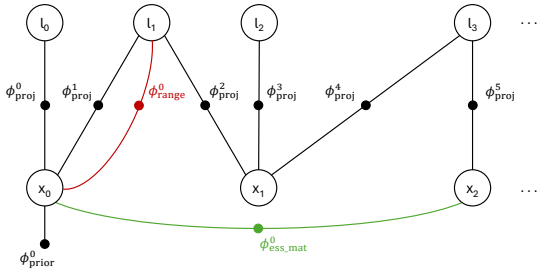


Figure 2: Illustration of the constructed factor graph for a subset of variables. l_1 is a LIDAR feature associated to both a projection factor and a range factor. The essential matrix factor is represented in green.

linear, iSAM2 requires initial values for all new variables added to the graph. While landmarks are initialized via

triangulation, there are two options to compute the poses' initial guess. The first relies on the P*n*P algorithm, which estimates the spacecraft pose with respect to an asteroid-fixed reference frame $\mathcal{A}$ using observations of already initialized landmarks [20]. This method is used when the inertial attitude of the asteroid is considered unknown. In contrast, the second method involves star tracker measurements for the camera attitude and the EKF prediction for the position. Since the filter estimates the spacecraft state in the inertial frame $\mathcal{N}$, the propagated position needs to be rotated to the asteroid-fixed frame before being used for the optimization process. Therefore, the knowledge of the target attitude $\mathbf{C}_{AN}$ (rotating from inertial to asteroid-fixed frame) is necessary, and is estimated by the additional algorithms described below (sections 3 and 4). Similarly, the orientation part of the initial guess corresponds to the attitude of the camera with respect to the asteroid-fixed frame. This can be obtained by combining star tracker measurements, simulated assuming Gaussian distributed errors, with the inertial orientation of the asteroid.

Loop closure detection is carried out using the classical bag of words (BoW) approach [21]. At each keyframe, up to 10000 ORB features are extracted and used to synthesize a global BoW descriptor. The latter is compared to those of previous images with a phasing of more than 10 frames, to compute a similarity score. The image with the highest score is considered as a candidate loop closure. A validation step based on feature matching attempt is then performed. Two new sets of ORB features are extracted from the candidate images and matched, which undergoes an outlier rejection step using MSAC combined with the direction of motion estimation algorithm described in Christian et al. [22]. If the process is successful and a minimum number of 10 inliers is found, a loop closure factor is added to the factor graph, in the form of an essential matrix constraint $\phi_{\text{ess\_mat}}$. The latter is constructed by combining the direction of motion estimate used to validate the loop closure with the orientation change between the two camera frames, obtained starting from star tracker measurements. This procedure increases accuracy and robustness with respect to directly estimating the full essential matrix from the candidate feature matches. After a loop closure is added, a further incremental optimization step is carried out. To initialize the algorithm, it is assumed that an initial state with associated uncertainty is available. Then, prior factors defined using prior knowledge and the EKF propagation are placed on all initial poses until the first set of landmarks is initialized.

2.3. Extended Kalman Filter design

Unlike the factor graph framework, the Extended Kalman Filter enables the estimation of the spacecraft's inertial position and velocity at any timestamp. The latter combines the position estimates produced by the graph solver, here treated as measurements, with a dynamical model of the spacecraft motion. The EKF propagation step is the first operation performed when a new keyframe is processed. The dynamics includes the asteroid central gravity and the SRP effect. Then, relevant factors are added to the graph and the optimization process is carried out. Finally, the EKF measurement update is run using the optimized spacecraft position as measurement. Specifically, the latest estimate of the position in the asteroid-fixed frame is retrieved from iSAM2 together with its marginal covariance. Both are converted to the inertial reference frame according to:

$$\vec{r}^{\mathcal{N}} = \mathbf{C}_{NA}\, \vec{r}^{\mathcal{A}} \tag{1}$$

$$P_{rr}^{\mathcal{N}} = \mathbf{C}_{NA}\, P_{rr}^{\mathcal{A}}\, \mathbf{C}_{NA}^{T} \tag{2}$$

Position measurements are slightly underweighted by inflating the covariance matrix before proceeding with the standard EKF update step, to mitigate the detrimental effect of neglecting correlations:

$$R = 4\, P_{rr}^{\mathcal{N}} \tag{3}$$

2.4. LIDAR Measurements Integration

LIDAR altimeter measurements are essential to break the scale ambiguity inherent in monocular cameras. Their integration in the SLAM system is based on the algorithm proposed in [23], but included here in a graph-based framework rather than a filtering context. The idea of the algorithm is to track specific features whose distance is known from the altimeter measurement. As the associated landmarks are optimized, the correct scale is provided by the LIDAR measurements and propagates to the rest of the solution.

The algorithm assumes that the LIDAR-camera cross-calibration is known. Thus, the LIDAR pointing direction can be directly associated to a pixel in the camera images. At each keyframe, a patch surrounding the coordinates of the LIDAR point is extracted from the image, as illustrated in fig. 3. Then, the corner response within that patch is computed using the Harris corner score formulation [16]. When the corner response is above a certain threshold, a LIDAR landmark is initialized and added to the factor graph. The initialization is straightforward, since the 3D point coordinates can be reconstructed by combining the image location with the altimeter measurement, without the need for triangulation. LIDAR features are inserted in the graph with two different factors: a projection factor that minimizes the reprojection error, and a range factor that constrains the depth of the landmark. After initialization, the LIDAR feature is tracked across subsequent frames. To increase robustness, an outlier rejection step is performed to verify that the tracked feature is consistent with the essential matrix estimated by the front-end. If the feature is successfully tracked until the next keyframe, a projection factor is added to the factor graph. It is highlighted that no additional range factors are added after the initialization step. Indeed, as the spacecraft moves and the asteroid rotates, the LIDAR measurement point moves across the surface and can no longer provide the range to the same feature. Nonetheless, if the LIDAR feature is successfully tracked for at least two keyframes, the initial measurement is enough

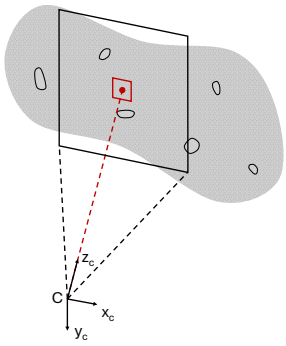


Figure 3: Illustration of the LIDAR feature patch, in red. The camera frame and FOV are represented in black. Image size not to scale.

to break scale ambiguity. More than one LIDAR feature can be tracked in parallel. However, to avoid redundancy, they are initialized only if the distance between the LIDAR pixel and the closest active LIDAR feature is larger than a threshold set equal to 1/4 of the largest image size. LIDAR measurements are simulated by adding white noise with a standard deviation of 10 m to the true range of the feature. Similarly, when depth factors are introduced in the graph, measurements are slightly underweighted by doubling the associated standard deviation.

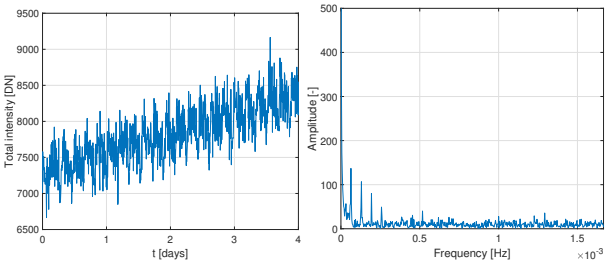
3. ROTATIONAL PERIOD ESTIMATION

The initial estimation of the asteroid rotational period is based on the observation of the light curve generated by the target at far distance, during the approach phase [24]. Each image is initially filtered to retain only pixels whose intensity is above a certain threshold. The sum of their intensities in every image generates a light curve, whose total intensity changes as the target rotates. An example over a 4-day time period is shown in fig. 4a for Bennu.

The light curve signal is then processed using the Fast Fourier Transform (FFT) algorithm to retrieve its frequency components. The periodicity induced by the rotation of the asteroid generates peaks associated to the rotational period and its multiples. In particular, the first two peaks of the FFT should be associated to the rotational period and its double. Therefore, if their ratio is close to 2, the estimate is considered valid and the period is retrieved from the associated frequency. Fig. 4b illustrates the FFT corresponding to the example light curve, showing the prominent peaks associated to the rotational period.

4. SPIN AXIS ESTIMATION

Knowledge of the target's spin axis is essential to define an asteroid-fixed reference frame and relate the surface-relative pose, estimated by the SLAM algorithm, with the inertial spacecraft state, estimated by the EKF. The Discrete Adaptive Angular Velocity Estimator (DAAVE) algorithm [25, 26] is used for this scope. The latter estimates the angular velocity direction given a sequence



(a) Light curve. (b) Single-sided FFT.

Figure 4: Example light curve and associated FFT for Bennu.

of quaternion measurements of an arbitrary frame over a sliding window. It is worth noting that the algorithm is also suitable for tumbling targets, although only asteroids in stable rotation are considered in this work.

More in detail, the inertial direction of the asteroid's angular velocity vector can be reconstructed by combining the target-relative attitude of the spacecraft, as provided by the SLAM algorithm, with the inertial estimates of the star tracker. Here, the SLAM system is run in a reduced mode without the need for the transformation between the inertial and the asteroid-fixed frame and without using EKF data. Similarly, LIDAR measurements are assumed to be unavailable given the larger distance to the target. Therefore, prior factors with small errors are placed on the initial poses to fix the scale of the solution, as is customary in these cases [27]. To construct the sequence of quaternions required by DAAVE, the arbitrary target-fixed frame is defined as the camera frame at the first image, denoted as $\mathcal{C}_0$. The SLAM algorithm provides the sequence of camera orientations with respect to the asteroid surface, $\vec{q}_{\mathcal{C}\backslash\mathcal{A}}(t)$. It is highlighted that, being $\mathcal{C}_0$ a target-fixed frame, $\vec{q}_{\mathcal{C}_0\backslash\mathcal{A}}$ is fixed in time, whereas $\vec{q}_{\mathcal{C}_0\backslash\mathcal{N}}(t)$ varies according to the asteroid rotation rate. Indicating quaternion composition with the convention $\boldsymbol{q}_{CA} = \boldsymbol{q}_{CB} \otimes \boldsymbol{q}_{BA}$, the time history of $\mathcal{C}_0$ with respect to the inertial frame can be obtained:

$$\vec{q}_{\mathcal{C}\backslash\mathcal{C}_0}(t) = \vec{q}_{\mathcal{C}\backslash\mathcal{A}}(t) \otimes \vec{q}_{\mathcal{A}\backslash\mathcal{C}_0} \tag{4}$$

$$\vec{q}_{\mathcal{C}_0\backslash\mathcal{N}}(t) = \vec{q}^{-1}_{\mathcal{C}\backslash\mathcal{C}_0}(t) \otimes \vec{q}_{\mathcal{C}\backslash\mathcal{N}}(t) \tag{5}$$

where $\vec{q}^{-1}_{\mathcal{B}\backslash\mathcal{A}} = \vec{q}_{\mathcal{A}\backslash\mathcal{B}}$ denotes the quaternion inverse. The quaternion measurements are aggregated into a 4xL matrix of the form:

$$Q_{\mathcal{C}_0\backslash\mathcal{N}} = [\vec{q}_{k-L+1} \quad \vec{q}_{k-L+2} \quad \cdots \quad \vec{q}_k] \tag{6}$$

where L is the size of the sliding window and for brevity $\vec{q}_k = \vec{q}_{\mathcal{C}_0\backslash\mathcal{N}}(t_k)$. The angular velocity direction can be found from the SVD of $Q_{\mathcal{C}_0\backslash\mathcal{N}}$ as detailed in [25, 26]. Finally, the $\vec{\omega}$ is further filtered by taking its average in time to obtain the final estimate, denoted $\hat{\vec{\omega}}$.

5. ANGULAR VELOCITY REFINEMENT

The light curve estimation method, discussed in section 3, provides an initial guess of the target's rotational period

that can be further refined during the intermediate distance phase. Inspired again by [26], the asteroid's angular velocity is estimated using a multiplicative EKF (MEKF) algorithm. This is initialized with the angular velocity magnitude and direction from the previous algorithms, and is run in parallel with spin axis estimation, using the estimates of the quaternion $q_{\mathcal{C}_0\backslash\mathcal{N}}(t)$ as measurements. The key idea behind this approach is that the MEKF is capable of refining the angular velocity estimate by enforcing the kinematics constraint as well as by weighting measurements according to their covariance, contrarily to DAAVE. The full derivation of the filter can be found in the reference paper [26].

6. RESULTS

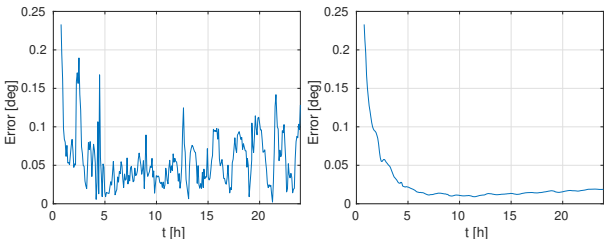
The presented set of algorithms is tested in a sequence of simulations, one per phase, each considering the previous results as initial input. Bennu is chosen as simulated test case scenario, being well-characterized but still challenging. Results are presented according to the order in which they are obtained: rotational period estimation; spin axis determination and angular velocity refinement; the full SLAM algorithm. A camera-in-the-loop image sequence obtained in the RETINA facility is also used to assess the last module [28]. A monocular camera with a resolution of 1024x1024 px and a FOV of 10x10 deg is considered for all simulations except the camera-in-the-loop test. For the intermediate and close distance phases, truth trajectories are generated considering the effect of the asteroid gravity field up to 4th order and degree, SRP (cannonball model), and the Sun's 3rd body perturbation.

6.1. Rotational Period Estimation

Rotational period estimation is carried out when the asteroid occupies only few pixels in the camera images. Simulations start at a distance of 250 km from the target. In this case, a simple rectilinear trajectory is simulated given the large distance and short time-span. The spacecraft velocity is thus assumed constant and equal to 0.05 m/s in the direction of the target. 4 days of simulation are considered, with an image acquired every 5 minutes. The approach phase angle starts at 34.3 deg, and the variation throughout the trajectory is in the order of 1 deg. The light curve and the associated FFT for the Bennu test case are reported in fig. 4, where the frequency peaks associated with the rotational period are clearly visible. The estimate produced by the algorithm is 4.2704 h, which corresponds to a relative error of 0.598%.

6.2. Spin Axis Estimation and Angular Velocity Refinement

This section presents the results of the spin axis and angular velocity estimation and refinement, with DAAVE and MEKF algorithms respectively. The initial uncertainty for the MEKF is set to 1% of the true value, while the measurement angular error is assumed equal to $\sigma_\theta = 0.5$ rad, and the process noise $Q_w = 10^{-22}$ rad^2/s^3. For the Bennu scenario, a hyperbolic trajectory similar to those



(a) Raw estimate error. (b) Averaged estimate error.

Figure 5: Bennu spin axis estimation error.

carried out in the characterization phase of typical small body missions is considered [29]. The hyperbolic arc is symmetric with respect to the Sun direction, and the distance is slightly below 6 km. The simulation has a duration of 1 day, with images acquired every minute. Both the raw estimate produced by the DAAVE algorithm and its averaged value are reported, run with an initial window size of 10 keyframes. After 5 hours the averaged estimate is found to closely match the true spin axis direction. This is confirmed by the angular errors computed between the true and estimated spin axis directions as shown in fig. 5. The averaged estimate is overall more stable, the raw estimation error being lower in the central phase, which corresponds to the part of the trajectory closest to the target. Even if the error starts to increase as the spacecraft moves away, an highly accurate final estimate is achieved, characterized by an error of 0.0186 deg. As for the angular spin rate, the refined estimate from the MEKF improves upon the initial guess, attaining an error of 0.0197 deg/day.

6.3. SLAM Algorithm - Simulated scenario

This section reports the results of the SLAM system, started from the previously obtained estimates. At the initial epoch t_0 the asteroid-fixed frame $\mathcal{A}_0$ is defined assuming the z axis aligned with the estimated spin axis. The x and y axes are chosen such to minimize the angle between the estimated and true directions, thus constructing $\hat{R}_{\mathcal{A}_k\backslash\mathcal{N}}$. Finally, the DCM $\hat{R}_{\mathcal{A}_k\backslash\mathcal{A}_0}$ accounting for the rotation of the asteroid between time instants t_0 and t_k is defined as:

$$\hat{R}_{\mathcal{A}_k\backslash\mathcal{A}_0} = \begin{bmatrix} \cos(\hat{\omega}\Delta t) & \sin(\hat{\omega}\Delta t) & 0 \\ -\sin(\hat{\omega}\Delta t) & \cos(\hat{\omega}\Delta t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (7)$$

where $\Delta t = t_k - t_0$. The attitude at any time t_k can be then computed by composing the two rotations. It is noted that the algorithm does not explicitly model the uncertainty on the asteroid attitude. The trajectory considered for Bennu is nearly circular, starting at a distance of 3 km, and lasts 1 day, with an image acquired every minute. The initialization error is assumed to be characterized by a Gaussian distribution with 0 mean and a standard deviation of 20 m for each position component, and 5 mm/s for the velocity. As for the frontend, the Harris corner detector is used. It is remarked that monocular SLAM solutions are defined up to a $Sim(3)$ transform.

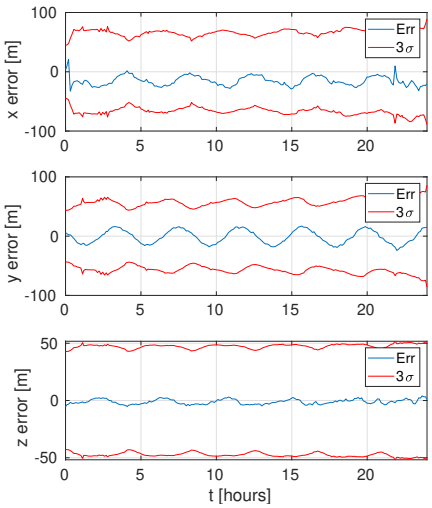


Figure 6: Position error of the smoothed solution for Bennu - Camera frame.

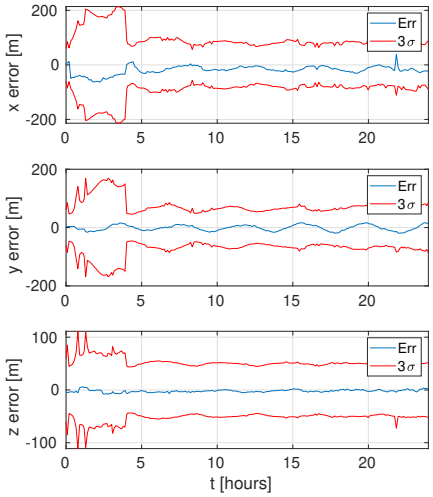


Figure 7: Position error of the instantaneous solution for Bennu - Camera frame.

Therefore, absolute rotation, translation, and scale are unobservable from images alone. In this work, the star tracker measurement fixes the absolute attitude while the LIDAR altimeter fixes scale, leaving only a global translation undetermined. Before evaluating errors, it is therefore necessary to register the reconstructed surface to the reference asteroid model, for instance using an ICP algorithm [30]. The estimated rigid translation is used to shift the map and the spacecraft trajectory, in order to align the CoM of the reference model with that of the map. This preserves all relative geometry and does not change the solution's validity, but enables the proper evaluation of the mapping errors in terms of navigation *relative* to the target. It is noted that the initial phase angle for the given trajectory is 34 deg, which increases up to more than 95 deg towards the end of the simulation, the latter being a challenging condition for vision-based algorithms. The position errors in the camera frame corresponding to the *smoothed* estimate are shown in fig. 6, i.e., the *best* estimate of the trajectory at the final time step, after all measurements have contributed to all states. The camera frame is defined with Z axis aligned to the boresight, and x, y axes to the image frame. The orientation error is defined as the local error between the true and estimated target-relative attitude. Plots of the latter are omitted for brevity. The results demonstrate good performance for all the considered quantities. For the position, the highest accuracy is observed in the z direction thanks to the highly effective LIDAR measurements. Errors in the x and y axes oscillate around 10-20 m of magnitude. Orientation is also well estimated, with error components below 0.5 deg throughout the simulation. Figure 7 reports the errors and associated uncertainty of the *instantaneous* estimate, corresponding to the best solution available after each keyframe update. Thus, the instantaneous values do not contain information introduced in the subsequent frames as the *smoothed* one, and can be interpreted as the best available estimate while running the algorithm online. The effect of loop closure is easily appreciable by considering the pose uncertainty. This grows significantly until the first successful loop closure is processed around 5 hours of simulation time, and settles to steady values soon after. In this scenario, the BoW approach is highly effective in correctly identifying loop closures. Due to the combination of the considered trajectory and the relatively fast rotation of the asteroid, the spacecraft re-observes surface regions from similar viewpoints. Furthermore, while illumination conditions change significantly across the entire simulation, their variations over one rotational period of the asteroid are more limited. These characteristics facilitate loop closure detections.

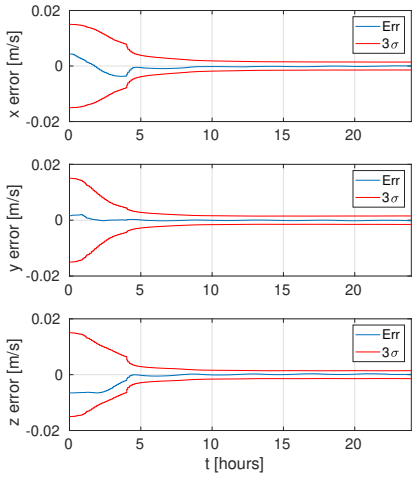


Figure 8: EKF velocity errors, Bennu test case, inertial frame.

As for the velocity, the EKF errors are reported in fig. 8, represented in the inertial frame. A steady convergence is observed in the initial part of the simulation, fostered by the first loop closure. After that, the error components settle around 0.1 mm/s, while the 3σ confidence bound remains around 1.5 mm/s. Position errors are only marginally better than the *instantaneous* SLAM solution, thanks to the contribution of the dynamical model. Fi-

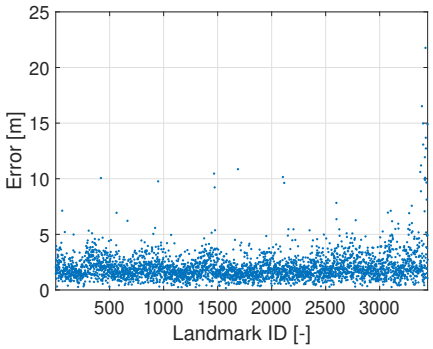


Figure 9: Map errors, Bennu test case.

nally, the mapping errors in fig. 9 confirms the accuracy of the navigation solution. The latter is computed as the minimum distance between each landmark and the closest point on the reference model. In most cases errors are below 5 m, with few instances around 10 m. Landmarks in the final part of the simulation show higher errors, as they have been observed in a small number of frames.

6.4. SLAM algorithm - Camera-in-the-loop

Finally, the results of a test with an image sequence acquired using the RETINA facility are presented. The same trajectory of the Bennu test case is used, shortened to 5 hours of simulation. As reported in figs. 10 to 12, the performance remains consistent with the synthetic test cases, despite different camera specifications, the presence of noise and image blur that were not considered before. The most noticeable difference is the lower number of LIDAR landmarks, mainly attributable to the image blur, which also decreases the corner response value.

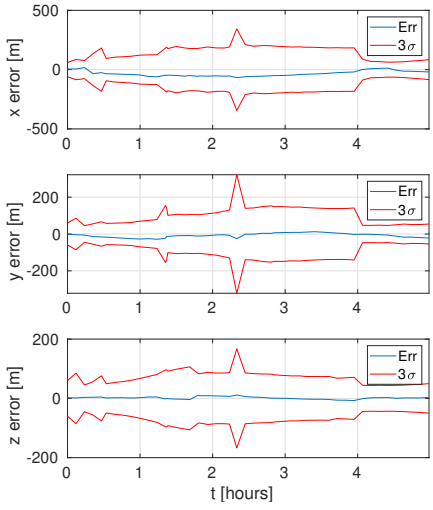


Figure 10: Instant. position error, RETINA test case.

7. CONCLUSION

This work shows that a single, hybrid pipeline can estimate an asteroid's spin state (period and axis), recover a

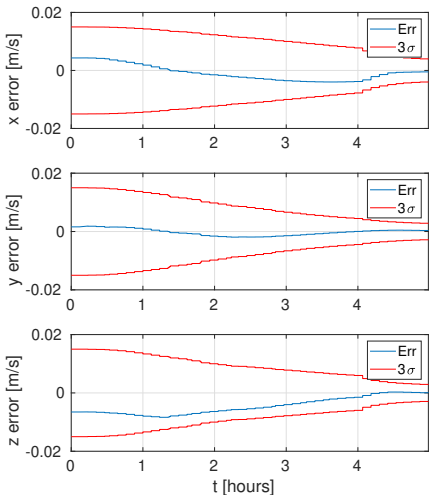


Figure 11: EKF velocity error, RETINA test case.

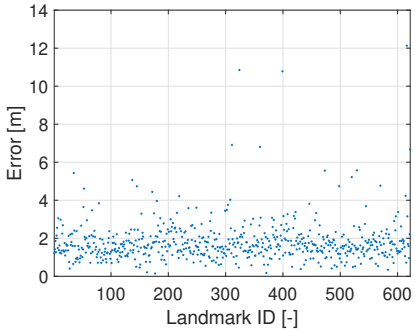


Figure 12: Map errors for the RETINA test case.

coarse shape, and localize a spacecraft during approach. The graph-based SLAM provides consistent mapping and pose, while a lightweight EKF supplies velocity without injecting full dynamics into the factor graph. LIDAR range points remove the monocular scale ambiguity with minimal added complexity. In simulation, the system achieves localization and mapping errors on the order of tens of meters. Loop closures, when available, sharply reduce pose uncertainty; where they do not occur, few LIDAR measurements along the radial direction still keep the solution well constrained. The presented set of algorithms may be extended to enable full on-board characterization by including centre of mass and gravitational parameter estimation, while additional visual measurements such as direction-of-motion may be added in the factor graph to improve accuracy. The incremental optimization was found to be sensitive to poor initial guesses. Robustness and computational efficiency on flight hardware remain key next steps, but the results indicate that the proposed approach is viable within realistic resource envelopes.

ACKNOWLEDGMENTS

This work is partially sponsored by the COSMICA project funded by the Italian Ministry of University and Research (MUR) through Decree No. R.0000398.24-10-2022.h.10:43. P.C. is sponsored by the European Union

under the Next Generation EU program, Mission 4 Component 1 (CUP D43C23002190008). P.P. and F.T. are sponsored by EXTREMA, a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant Agreement No.864697).

REFERENCES

[1] Bernard Godard et al. Multi-arc orbit determination to determine rosetta trajectory and 67p physical parameters. In *Proceedings 26th International Symposium on Space Flight Dynamics–26th ISSFD, Matsuyama, Japan*, 2017.

[2] Jason M Leonard et al. Osiris-rex orbit determination performance during the navigation campaign. In *2019 AAS/AIAA Astrodynamics Specialist Conference*, number AAS 19-714, 2019.

[3] RW Gaskell et al. Characterizing and navigating small bodies with imaging data. *Meteoritics & Planetary Science*, 43(6):1049–1061, 2008.

[4] Cedric Cocaud et al. Slam-based navigation scheme for pinpoint landing on small celestial body. *Advanced Robotics*, 26(15):1747–1770, 2012.

[5] David Nakath et al. Active asteroid-slam. *Journal of Intelligent & Robotic Systems*, 99(2):303–333, 2020. doi: 10.1007/s10846-019-01103-0.

[6] Paolo Panicucci. *Autonomous vision-based navigation and shape reconstruction of an unknown asteroid during approach phase*. PhD thesis, Toulouse, ISAE, 2021.

[7] Brent E Tweddle et al. Factor graph modeling of rigid-body dynamics for localization, mapping, and parameter estimation of a spinning object in space. *Journal of Field Robotics*, 32(6):897–933, 2015.

[8] Timothy P Setterfield et al. Mapping and determining the center of mass of a rotating object using a moving observer. *The International Journal of Robotics Research*, 37(1):83–103, 2018.

[9] Kaitlin Dennison et al. Autonomous asteroid characterization through nanosatellite swarming. *IEEE Transactions on Aerospace and Electronic Systems*, 59(4):4604–4624, 2023. doi: 10.1109/TAES.2023.3245997.

[10] Mehregan Dor et al. Astroslam: Autonomous monocular navigation in the vicinity of a celestial small body — theory and experiments. *The International Journal of Robotics Research*, 43(11):1770–1808, 2024. doi: 10.1177/02783649241234367.

[11] Pietro Califano et al. Enhanced full visual slam for close proximity exploration of asteroids combining incremental smoothing with filtering techniques. In *35th AAS/AIAA Space Flight Mechanics Meeting*, pages 1–22, 2025.

[12] M Cardi et al. The hera milani mission. In *Small Satellites Systems and Services Symposium (4S 2024)*, volume 13546, pages 677–691. SPIE, 2025.

[13] Jacopo Villa et al. Optical navigation for autonomous approach of small unknown bodies. In *Proceedings of the 43rd Annual AAS Guidance, Navigation & Control Conference*, pages 13–17, Maui, HI, USA, 2019.

[14] Michael Kaess et al. isam2: Incremental smoothing and mapping using the bayes tree. *The International Journal of Robotics Research*, 31(2):216–235, 2012.

[15] Jean-Yves Bouguet. Pyramidal implementation of the lucas kanade feature tracker description of the algorithm. *Intel Corporation*, 2001.

[16] Chris Harris et al. A combined corner and edge detector. In *Alvey Vision Conference*, Manchester, UK, 1988.

[17] Benjamin J Morrell et al. Autonomous feature tracking for autonomous approach to a small body. In *ASCEND 2020*, Nov. 2020. doi: 10.2514/6.2020-4151.

[18] Richard Hartley et al. *Multiple View Geometry in Computer Vision*, page 287. Cambridge University Press, 2003.

[19] S. Nousias et al. Large-scale, metric structure from motion for unordered light fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3287–3296, 2019.

[20] Sébastien Henry et al. Optimal dlt-based solutions for the perspective-n-point. *arXiv preprint arXiv:2410.14164*, 2024.

[21] Dorian Gálvez-López et al. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on robotics*, 28(5):1188–1197, 2012.

[22] John A Christian et al. Image-based lunar terrain relative navigation without a map: Measurements. *Journal of Spacecraft and Rockets*, 58(1):164–181, 2021. doi: 10.2514/1.A34875.

[23] Ivan Alberico et al. Structure-invariant range-visual-inertial odometry. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10613–10620. IEEE, 2024.

[24] Santarshi Bandyonadhyay et al. Silhouette-based 3d shape reconstruction of a small body from a spacecraft. In *2019 IEEE Aerospace Conference*, pages 1–13. IEEE, 2019.

[25] Daniele Mortari et al. Discrete and continuous time adaptive angular velocity estimators. In *Proceedings of the AAS/AIAA Space Flight Mechanics Conference, Williamsburg, VA*, 2015.

[26] Marcelino M de Almeida et al. Real-time angular velocity estimation of non-cooperative space objects using camera measurements. In *AAS/AIAA Astrodynamics Specialist Conference*, volume 8, 2018.

[27] Mehregan Dor et al. Visual slam for asteroid relative navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2066–2075, 2021.

[28] Paolo Panicucci et al. Design of a low-aberration variable-magnification optical stimulator for vision system hardware-in-the-loop testing. In *IEEE Transactions on Aerospace and Electronic Systems*, pages 1–22, 2025. doi: 10.1109/TAES.2025.3580393.

[29] Fabio Ferrari et al. Trajectory options for hera's milani cubesat around (65803) didymos. *The Journal of the Astronautical Sciences*, 68(4):973–994, 2021.

[30] Paul J Besl et al. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. Spie, 1992.